Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A multipurpose audio watermarking algorithm with synchronization and encryption

Baiying LEI[†], Ing Yann SOON

(*School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*)

[†]E-mail: leib0001@e.ntu.edu.sg

Received Apr. 6, 2011; Revision accepted Sept. 9, 2011; Crosschecked Dec. 8, 2011

Abstract: We propose a new multipurpose audio watermarking scheme in which two complementary watermarks are used. For audio copyright protection, the watermark data with copyright information or signature are first encrypted by Arnold transformation. Then the watermark data are inserted in the low frequency largest significant discrete cosine transform (DCT) coefficients to obtain robustness performance. For audio authentication, a chaotic signal is inserted in the high frequency insignificant DCT coefficients to detect tampered regions. Furthermore, the synchronization code is embedded in the audio statistical characteristics to resist desynchronization attacks. Experimental results show that our proposed method can not only obtain satisfactory detection and tampered location, but also achieve imperceptibility and robustness to common signal processing attacks, such as cropping, shifting, and time scale modification (TSM). Comparison results show that our method outperforms some existing methods.

Key words: Audio watermarking, Copyright protection, Content authentication, Compound encryption, Synchronization
doi:10.1631/jzus.C1100085   **Document code:** A   **CLC number:** TP391; TP309

1 Introduction

With the rapid development of the Internet, attention and concerns on intellectual property protection have been raised in recent years. As a result, among many others, a digital watermarking method has been proposed to prevent the unauthorized copying and distribution of multimedia data by inserting ‘undetectable’ copyright information into the host signal without much signal distortion. This method effectively addresses the data security and authentication issues. In general, a successful audio watermarking algorithm should be robust to temporal scaling and resist most common signal processing manipulations while the signal-to-noise ratio (SNR) should be greater than 20 dB, which is specified in the International Federation of the Phonographic Industry (IFPI), STEP2000, and Secure Digital Music Initiative (SDMI) (Katzenbeisser and Petitcolas, 2000).

To this end, many audio watermarking schemes have been proposed. Swanson *et al*. (1998) proposed a robust audio watermarking scheme by adopting perceptual masking to increase the robustness. Kirovski and Malvar (2003) recommended the use of a spread spectrum method for audio copyright protection. Yeo and Kim (2003) proposed a patchwork method to protect audio copyright. Cvejic and Seppanen (2008) gave a general review of recent watermarking techniques and audio watermarking algorithms. Interested readers can refer to Cvejic and Seppanen (2008) for further understanding.

Recently, another trend has been witnessed. Rather than achieving copyright protection or content authentication separately in different watermarking systems, many multipurpose watermarking techniques have been proposed to achieve these goals at the same time (Lu CS *et al*., 2000; Lu and Liao, 2001; Lu ZM *et al*., 2005; Chen and Zhu, 2008; Ma *et al*., 2008). For example, Lu *et al*. (2000) first proposed a cocktail watermarking scheme for audio protection and authentication. This was realized by quantizing audio's fast Fourier transform coefficients as masking threshold units. This scheme uses the human auditory system to improve robustness, but the results after attacks are not quite satisfactory. Chen and Zhu (2008) designed a multipurpose audio watermarking algorithm with a combination of different techniques like discrete wavelet transform (DWT), discrete cosine transform (DCT), independent component analysis, and vector quantization, and achieved a satisfactory performance to some extent. However, the desynchronization attacks, including shifting, cropping, and time scale modification (TSM) attacks, were not reported in the above mentioned multipurpose schemes. Actually, desynchronization attacks have been, for a long time, a challenging problem. To tackle this issue, several synchronous audio watermarking schemes were proposed (Huang *et al*., 2002; Wang and Zhao, 2006; Zhang *et al*., 2006). Indeed, these audio watermarking methods solve the synchronization problems and are robust to cropping, shifting, and TSM attacks to some extent. However, most of these systems do not consider the combination of other features such as chaotic encryption to achieve more robust results and the multipurpose target.

As is well known, encryption can be a useful tool for improving watermarking security. Thus, it is very popular in the digital watermarking field and there are many papers describing integrated encryption and watermarking (Xie *et al*., 2006; Wu and Shih, 2007). In this paper, a new multipurpose audio watermarking technique is developed. Different from most of the existing chaotic watermarking schemes, two chaotic maps are employed in our scheme to enhance watermarking confidentiality. A 2D Arnold cat map is adopted to scramble the embedding position. A watermark (binary image signal) is shuffled randomly in the whole host audio to ensure the security of the watermarking algorithm. Meanwhile, another chaotic map, the Henon map, is employed to generate the watermark signal for tampered localization and embedded in the fragile watermarking scheme.

2 Chaotic maps and watermarking

Chaotic signals, generated by simple dynamic systems, have the behavior of certain nonlinear dynamic systems (Tsekeridou *et al*., 2001). Chaos can produce a number of uncorrelated, random-like, yet deterministic chaotic signals with small perturbation of parameters. As a chaotic signal is complicated and very difficult to predict over a long time, it has been widely used in the watermarking and encryption field. Besides, chaotic signals can be generated very easily by specifying the initial conditions and chaotic parameters. Actually, a 1D chaotic map is used to generate a 1D sequence of real numbers and is usually defined as

$$x(n+1)=f(x(n),\lambda), \tag{1}$$

where n is the map iteration index, n=1, 2, …, and λ is the system parameter.

To scramble the embedding position of the watermark image, 2D cat mapping, also called the Arnold transformation, is exploited to map one matrix into another. Arnold transformation can be regarded as a discrete chaotic system. In the field of watermarking, Arnold transformation is very popular and has been widely used as a method to shuffle the image. The basic principle of cat mapping is to scramble pixel position within the image. The generalized 2D invertible chaotic map is usually defined as

$$\begin{aligned}\begin{bmatrix} x(n+1) \\ y(n+1) \end{bmatrix} &= \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x(n) \\ y(n) \end{bmatrix} \bmod N \\ &= \boldsymbol{P}\begin{bmatrix} x(n) \\ y(n) \end{bmatrix} \bmod N,\end{aligned} \tag{2}$$

where a, b, c, d are positive integers. To guarantee that the Arnold transformation is a one-to-one mapping, $\boldsymbol{P}$ should satisfy $|\boldsymbol{P}|=ad-bc=1$. In this case, only three of four parameters are independent. Consequently, the iteration number n, the parameters a, b, c, and the initial values are employed as the encryption keys. In our scheme, we set a=2, b=c=d=1. The encryption goal can be achieved by shuffling the watermark positions. Using the generalized cat map, the watermark embedding position can be obtained. Therefore, the position of the encrypted watermark is determined using the Arnold transformation, which can enhance watermark security. In our algorithm, the watermark data are first mapped using the Arnold transformation, and then reshaped to 1D data $w(i)$ for robust watermark embedding.

Another chaotic map, the Henon map, is used to randomly select the embedding points. The generalized Henon map is defined as

$$z(n)=1+B(z(n-2)-z(n-3))+Cz^2(n-2), \quad (3)$$

where B and C are selected as B=0.3 and 1.07≤C≤ 1.09. The initial values are chosen in the range of (−1.5, 1.5) to ensure that the chaotic systems are in the chaotic states and have chaotic attractors. Modulo operation is usually performed to restrict the chaotic sequences within limits and in the chaotic states. Finally, chaotic sequence $z(n)$ is transformed into a binary sequence ww(i) with a specified threshold and used in fragile watermark embedding.

3 Methodology

3.1 System diagram

In our proposed audio watermarking scheme, the main idea is to divide the original host audio into two portions. In one portion, the synchronization code is inserted in the time domain. In the other portion, the robust and fragile watermarks are embedded into the DCT coefficients in the low- and high-frequency subbands, respectively. The synchronization code insertion and watermark embedding procedure is summarized in Fig. 1.

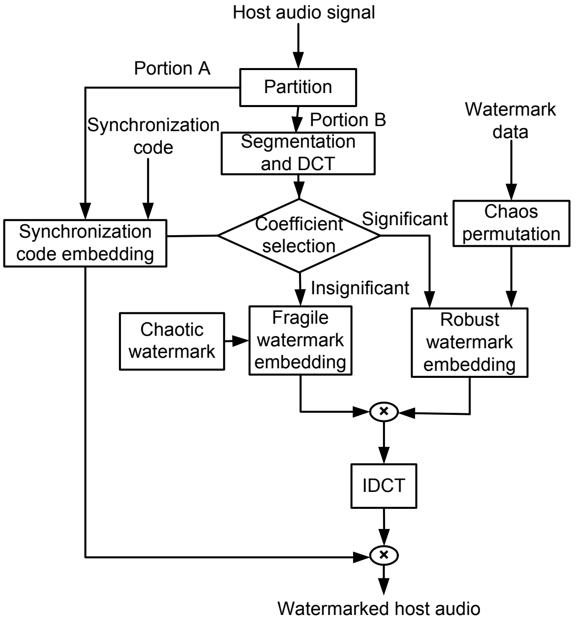

Fig. 1 Diagram of the watermark embedding procedure

3.2 Synchronization code technique

In the audio watermarking field, several watermarking schemes exploit the synchronization code (Huang *et al.*, 2002; Wang and Zhao, 2006; Wang and Yang, 2008). Synchronization code is used to locate the positions of hidden informative bits in order to resist the cropping and shifting attacks. Synchronization code can be inserted in the time or frequency domain. In fact, desynchronization attacks may cause a serious problem to any watermarking scheme. After such attacks as cropping, shifting, and MP3 compression, we usually fail to extract the watermarks. Therefore, the correct position of the watermark must be identified or localized before extraction. The synchronization problem can be addressed by integrating a synchronization code with watermark bits to reproduce a binary sequence. The synchronization code is often put in front of the watermark for identifying the position of the inserted watermark. Fig. 2 shows the data structure of this method. The cover signal is divided into proper segments according to the length of Syn(k) and W(k). Owing to a lower computational cost, the synchronization code is inserted in the time domain and forms a binary sequence.

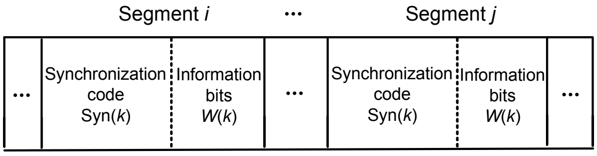

Fig. 2 Data structure of the hidden bitstream

One portion of the audio signal S, used to embed the synchronization code, is split into proper segments according to the length of the synchronization code. In our method, the portion A signal is divided into $L_{\rm A}$ audio segments with n samples, denoted as

$$S_{\rm A}(k)(j)=S(kn+j), \quad 1\le k\le L_{\rm A},\ 1\le j\le n. \quad (4)$$

Our algorithm exploits a chaotic sequence as the synchronization code in front of the watermark, to locate the position where the watermark is embedded. A logistic chaotic sequence in interval [0, 1] is used to generate the synchronization code:

$$y(k+1)=\lambda y(k)(1-y(k)), \quad 1\le k\le L_{\rm A}, \quad (5)$$

where 3.57<λ≤4. Let T_{s} be the synchronization code threshold. In our scheme, T_{s}=0.5. Then $y(k)$ is transformed into the synchronization sequence Syn(k) with the following rule:

$$\mathrm{Syn}(k)=\begin{cases}1, & y(k)>T_{\mathrm{s}},\\ 0, & \text{otherwise.}\end{cases} \tag{6}$$

According to statistics, a slight change of the audio signal may not greatly modify the statistical average. To achieve a tradeoff between inaudibility and robustness of the watermarking algorithm, the synchronization code is embedded into the statistical average of audio samples. Thus, the mean value of each segment, $S_{\mathrm{A}}(k)$, is first calculated by

$$\overline{S_{\mathrm{A}}(k)}=\frac{1}{n}\sum_{j=1}^{n}S_{\mathrm{A}}(k)(j),\quad 1\le k\le L_{\mathrm{A}}. \tag{7}$$

Then the synchronization code is inserted into $S_{\mathrm{A}}(k)$ bit by bit as follows:

$$S'_{\mathrm{A}}(k)(j)=\begin{cases}S_{\mathrm{A}}(k)(j)+2\left|\overline{S_{\mathrm{A}}(k)}\right|, & \mathrm{Syn}(k)=1,\\ S_{\mathrm{A}}(k)(j)-2\left|\overline{S_{\mathrm{A}}(k)}\right|, & \mathrm{Syn}(k)=0.\end{cases} \tag{8}$$

The reason for adding and subtracting the mean value is that the mean value is made positive when bit '1' is embedded and negative when '0' is embedded. After the synchronization code is inserted, we need to extract and detect the synchronization code. Supposing $S''_{\mathrm{A}}(k)$ is the embedded and attacked signal, the mean value of each segment $\overline{S''_{\mathrm{A}}(k)}$ is calculated as

$$\overline{S''_{\mathrm{A}}(k)}=\frac{1}{n}\sum_{j=1}^{n}S''_{\mathrm{A}}(k)(j),\quad 1\le k\le L_{\mathrm{A}}. \tag{9}$$

In the synchronization code extraction procedure, the audio samples also follow the rules in the embedding process, and the mean value of each segment is calculated. The extraction is based on the popular odd-even parity rule:

$$\mathrm{Syn}'(k)=\begin{cases}1, & \overline{S''_{\mathrm{A}}(k)}\ge 0,\\ 0, & \overline{S''_{\mathrm{A}}(k)}<0.\end{cases} \tag{10}$$

Finally, we use the normalized correlation coefficient (NC) to calculate the similarity between the original and extracted synchronization codes:

$$\mathrm{NC}(\mathrm{Syn},\mathrm{Syn}')=\frac{\sum_{k=1}^{n}\mathrm{Syn}'(k)\cdot\mathrm{Syn}(k)}{\sqrt{\sum_{k=1}^{n}\mathrm{Syn}(k)^2}\sqrt{\sum_{k=1}^{n}\mathrm{Syn}'(k)^2}}, \tag{11}$$

where Syn(k) and Syn′(k) are the original and extracted synchronization codes, respectively. If the NC between Syn(k) and Syn′(k) is greater than or equal to a specified threshold T, then Syn′(k) is considered a synchronization code; otherwise, no synchronization code is detected or extracted.

3.3 Watermark embedding

As shown in Fig. 1, the second portion of the host audio signal is used to embed the robust and fragile watermarks according to coefficient selection rules. The embedding process contains the following steps.

Step 1: Divide audio clips into non-overlapping segments.

Step 2: Perform DCT on each segment to obtain a set of coefficients SS(i), i=1, 2, …, M×N, where M and N are the watermark length and width, respectively.

Step 3: After DCT transform, SS(i) is divided into significant part S_{B} and insignificant part S_{C} according to the coefficient selection rule. The most significant part and the largest coefficients in the low-frequency subband are selected to embed the robust watermark, while the insignificant part in the high-frequency domain is used for fragile watermarking.

For robust watermark embedding, the embedding rules are as follows:

If $w(i)$=1, then

$$S'_{\mathrm{B}}(i)=\begin{cases}\left\lfloor\frac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor\varDelta, & \mathrm{mod}(\left\lfloor\frac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor,2)=0,\\ \left\lfloor\frac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor\varDelta+\varDelta, & \mathrm{mod}(\left\lfloor\frac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor,2)=1,\ S_{\mathrm{B}}(i)\ge 0,\\ \left\lfloor\frac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor\varDelta-\varDelta, & \mathrm{mod}(\left\lfloor\frac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor,2)=1,\ S_{\mathrm{B}}(i)<0,\end{cases} \tag{12}$$

where $\varDelta$ denotes the quantization step and $\lfloor x\rfloor$ means the largest integer less than or equal to x.

If $w(i)$=0, then

$$S'_{\mathrm{B}}(i)=\begin{cases}\left\lfloor \dfrac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor \varDelta, & \mathrm{mod}(\left\lfloor \dfrac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor,2)=1,\\ \left\lfloor \dfrac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor \varDelta+\varDelta, & \mathrm{mod}(\left\lfloor \dfrac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor,2)=0,\ S_{\mathrm{B}}(i)\geq 0,\\ \left\lfloor \dfrac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor \varDelta-\varDelta, & \mathrm{mod}(\left\lfloor \dfrac{S_{\mathrm{B}}(i)}{\varDelta}\right\rfloor,2)=0,\ S_{\mathrm{B}}(i)<0.\end{cases} \tag{13}$$

For fragile watermarking, the simple SS method is used, which is denoted by

$$S'_{\mathrm{C}}(i)=S_{\mathrm{C}}(i)+\alpha\cdot \mathrm{ww}(i), \tag{14}$$

where α is the fragile watermark strength factor.

Step 4: Combine the two parts and apply inverse DCT to obtain the watermarked audio.

3.4 Watermark extraction and detection

In the watermark extraction process, the synchronization code should be searched for and extracted first based on Eq. (10). Then the robust watermark is extracted and the fragile watermark is detected using the same procedure as in the watermark embedding process. After watermarked audio segmentation and DCT, $S''_{\mathrm{B}}(i)$ and $S''_{\mathrm{C}}(i)$ are obtained. The detailed extraction process is as follows.

For robust watermark extraction:

If $\mathrm{mod}(\lfloor S''_{\mathrm{B}}(i)/\varDelta\rfloor,2)=1$, then the watermark is extracted using

$$w'(i)=\begin{cases}0, & -\dfrac{\varDelta}{2}\leq S''_{\mathrm{B}}(i)-\left\lfloor \dfrac{S''_{\mathrm{B}}(i)}{\varDelta}\right\rfloor \leq \dfrac{\varDelta}{2},\\ 1, & \text{otherwise.}\end{cases} \tag{15}$$

If $\mathrm{mod}(\lfloor S''_{\mathrm{B}}(i)/\varDelta\rfloor,2)=0$, then the watermark is extracted using

$$w'(i)=\begin{cases}1, & -\dfrac{\varDelta}{2}\leq S''_{\mathrm{B}}(i)-\left\lfloor \dfrac{S''_{\mathrm{B}}(i)}{\varDelta}\right\rfloor \leq \dfrac{\varDelta}{2},\\ 0, & \text{otherwise.}\end{cases} \tag{16}$$

For the fragile watermarking scheme, the detection mechanism is

$$\mathrm{ww}'(i)=(S''_{\mathrm{C}}(i)-\mathrm{S}'_{\mathrm{C}}(i))/\alpha. \tag{17}$$

After extraction, $w'(i)$ is reshaped to 2D, and then decrypted by cat mapping to obtain the extracted watermark, while the fragile extracted 1D watermark $\mathrm{ww}'(i)$ does not need any decryption.

4 Experimental results

Computer simulations were undertaken to evaluate the performance of our proposed method using the criteria of SNR, bit error ratio (BER), and NC. Speech and music signals (mono, 16 bits/sample, 44.1 kHz, WAV format) were used to conduct performance analysis. Classic music signals dlg.wav and piano.wav were used for performance comparison. The duration of the audio signal was 32 s. The embedded watermark was the binary logo image of size 32×32=1024 bits. The 16-bit synchronization code was ‘1111100110101110’ (or ‘0000011001010001’), and an audio segment had a length of n=1024 samples for embedding the synchronization code. We set α=0.2 and $\varDelta$=0.1. All these parameters were chosen to achieve a good compromise between the conflicting requirements of imperceptibility, robustness, and payload. The threshold T was set as 0.9.

4.1 Results of fragile watermarking

In the tampered localization performance test, we used four different tampered positions in the watermarked audio: (1) Samples 1000–6000 were replaced with zero; (2) Samples 30 000–36 000 were replaced with 0.1; (3) Samples 13 000–19 000 were replaced with other audio clips of the same length; (4) Samples 62 000–69 000 were changed with samples 70 000–77 000 of the same audio. Fig. 3 shows the

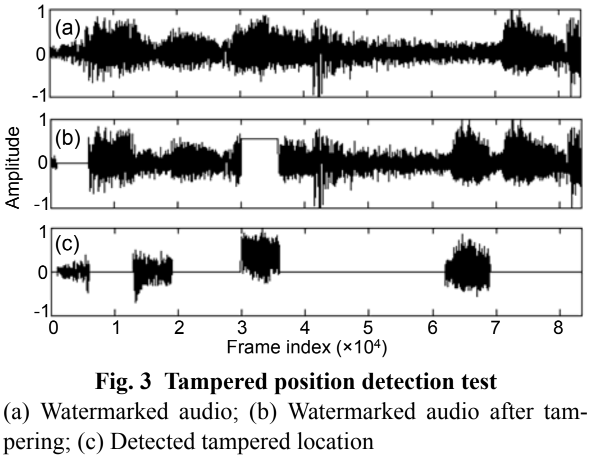

Fig. 3 Tampered position detection test

(a) Watermarked audio; (b) Watermarked audio after tampering; (c) Detected tampered location tampered position detection results. The proposed fragile watermarking scheme can detect the tampered positions with high accuracy, as the four different tampered locations were detected correctly with reliable detection ratios.

The tampered localization performance was evaluated after the following common signal processing manipulations: (a) MP3 compression attacks (48 kb/s); (b) low-pass filtering (LPF) (with a cut-off frequency of 22.05 kHz); (c) requantization test (16–8–16 bits); (d) white noise addition (2 dB); (e) delay (500 ms, 10%); (f) echo addition (500 ms, 10%); (g) resampling (44.1–22.05–44.1 kHz); (h) cropping (10% samples were set to zero); (i) shifting (by 10%). The NC results after common signal processing attacks are plotted in Fig. 4. The proposed fragile watermarking is more sensitive to common signal processing manipulations like LPF, white noise, delaying, and resampling attacks than the algorithm of Chen and Zhu (2008). The results of 0.589, 0.58, 0.59, and 0.59 after the LPF, white noise, delaying, and resampling, respectively, are higher than that of our scheme. However, the NC results of our proposed fragile watermarking scheme are relatively high after cropping and shifting synchronization attacks, as we adopt the synchronization technique to resist these attacks. Therefore, the performance of fragile watermarking after shifting and cropping attacks is not satisfactory.

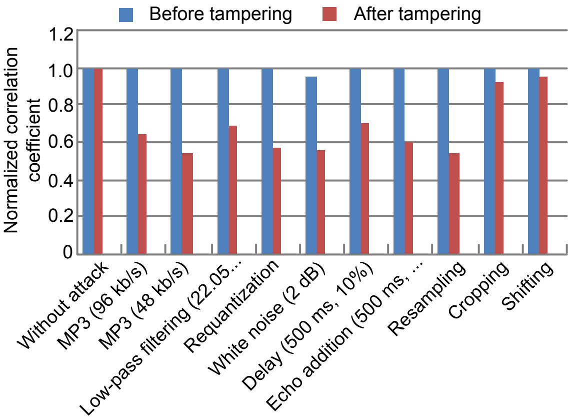

Fig. 4 Fragile watermarking performance after common signal processing attacks

4.2 Results of robust watermarking

4.2.1 Imperceptibility test

A transparency test was conducted with 100 different audio pieces of music signal with different properties and measured in terms of SNR, which is computed as follows:

$$\mathrm{SNR}=10\lg\left(\sum_{t=1}^{L}S^2(t)\bigg/\sum_{t=1}^{L}(S'(t)-S(t))^2\right), \quad (18)$$

where $S'(t)$ and $S(t)$ are the numbers of watermarked and original audio pieces, respectively.

Fig. 5 shows the SNR of 100 original host audio pieces and watermarked audio pieces using our proposed algorithm and the algorithm proposed by Chen and Zhu (2008). The average SNR using our method is 62.2 dB, while the average SNR using Chen and Zhu (2008)'s method is 40.4 dB. According to IFPI, SNR should be higher than 20 dB. Our proposed scheme totally satisfies this requirement.

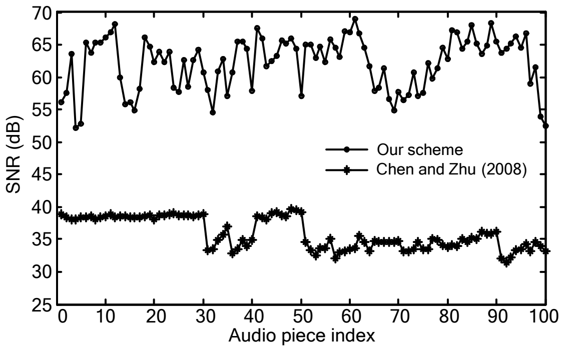

Fig. 5 Signal-to-noise ratios (SNR) of 100 original host audio pieces and watermarked audio pieces using our algorithm and the algorithm proposed by Chen and Zhu (2008)

4.2.2 Robustness to common attacks

Experimental results regarding desynchronization attacks and common audio signal processing manipulations are reported to evaluate the robustness of our proposed scheme. Typical types of desynchronization attacks such as random cropping and shifting were tested. As A/D and D/A conversions are not easy to conduct and are not very commonly used in the robustness test, we do not report the A/D or D/A attack here. We added the following attacks in our robustness tests: (j) TSM (−2%); (k) TSM (+2%); (l) TSM (−4%); (m) TSM (+4%). Table 1 summarizes the NC values of the robustness performance of attacks (a)–(m) using our method and the algorithms of Lu *et al.* (2000) and Chen and Zhu (2008). Our proposed watermarking is very robust to the common signal processing attacks. The robustness performance is similar to that of the scheme in Chen and Zhu (2008) and much better than that of the scheme in Lu *et al.* (2000). The desynchronization attack such as TSM is not a problem in our scheme as our scheme is robust to these kinds of attacks. The extracted watermarks shown in Fig. 6 suffering these attacks further confirm the robustness of our scheme, as there is almost no distortion due to very high NC values between the extracted watermarks and the original watermarks. The extracted watermarks suffering echo addition and delay attacks, however, are not quite satisfactory. Additional techniques are needed to further improve the performance.

Table 1 Robustness results under common signal processing attacks

Attack	Normalized correlation coefficient		
	Chen and Zhu (2008)	Lu *et al.* (2000)	Ours
No	1	1	1
(a)	1	0.726	1
(b)	1	0.833	1
(c)	0.998	0.961	0.999
(d)	0.923	0.963	0.905
(e)	1	N/A	0.974
(f)	0.996	N/A	0.956
(g)	0.998	0.988	1
(h)	N/A	N/A	1
(i)	0.651	N/A	1
(j)	N/A	N/A	1
(k)	N/A	N/A	1
(l)	N/A	N/A	1
(m)	N/A	N/A	1

N/A: not reported in the reference. (a) MP3 compression attacks (48 kb/s); (b) low-pass filtering (LPF) (with a cut-off frequency of 22.05 kHz); (c) requantization test (16–8–16 bits); (d) white noise addition (2 dB); (e) delay (500 ms, 10%); (f) echo addition (500 ms, 10%); (g) resampling (44.1–22.05–44.1 kHz); (h) cropping (10% samples were set to zero); (i) shifting (by 10%); (j) TSM (−2%); (k) TSM (+2%); (l) TSM (−4%); (m) TSM (+4%)

4.2.3 Robustness to MP3 compression

We tested the robustness against MP3 compression at various bitrates, and compared it with that of the scheme in Huang *et al.* (2002). Table 2 shows the MP3 compression and comparison results. Our scheme outperforms the scheme in Huang *et al.* (2002) and is very robust to the MP3 compression attack. Synchronization codes can be detected correctly even when the SNR values are very low (about 15 dB for MP3 compression) after being attacked.

Table 2 Robustness to MP3 compression*

Bitrate (kb/s)	SNR (dB)			
	dlg.wav		piano.wav	
	Huang *et al.* (2002)	Ours	Huang *et al.* (2002)	Ours
256	25.7	37.3	18.0	26.9
128	25.2	36.2	18.0	24.3
64	17.2	33.4	17.5	22.3
48	11.4	25.8	16.0	18.7
32	6.2	18.6	12.9	15.3

* In all cases, BER=0

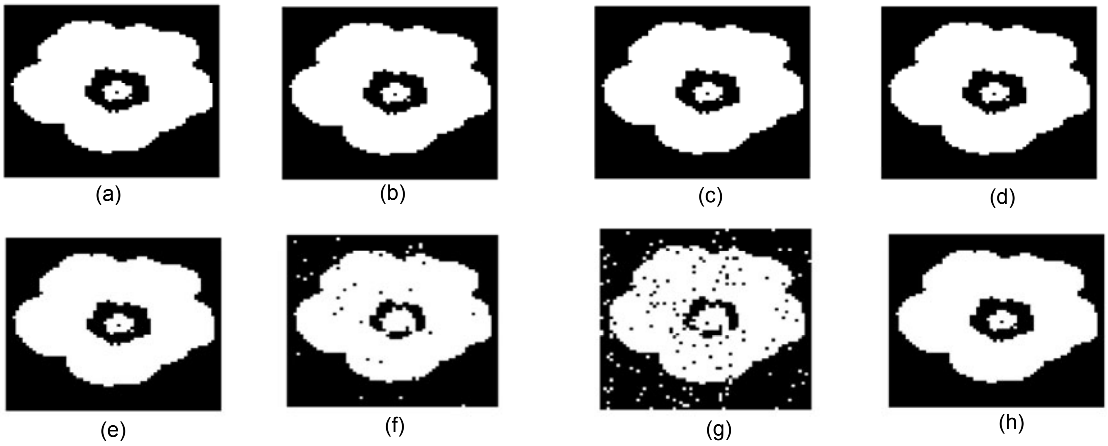

Fig. 6 Robust watermarks extracted under common audio signal processing

(a) No attack; (b) MP3 compression (64 kb/s); (c) Low-pass filtering (with a cut-off frequency of 22.05 kHz); (d) Requantization; (e) White noise addition (2 dB); (f) Delay (500 ms, 10%); (g) Echo addition (500 ms, 10%); (h) Resampling

4.2.4 Robustness of synchronization codes

In general, in the process of finding the synchronization code, false detection should be avoided to improve the robustness. In fact, the code length and the probability of ‘0’ and ‘1’ in the synchronization code determine whether or not we can find the synchronization code. Usually, longer synchronization codes indicate a better robustness of the scheme. In our scheme, the robustness of the synchronization codes was tested against various attacks. We also compared our synchronization code detection results with those using the schemes in Huang *et al.* (2002) and Wang and Zhao (2006). It is found that our scheme and the scheme in Wang and Zhao (2006) can both find the synchronization codes correctly under all mentioned attacks. In contrast, the scheme in Huang *et al.* (2002) cannot locate the synchronization codes under such attacks as resampling and very low bitrate MP3 compression.

4.2.5 Benchmark test

To further evaluate the robustness performance of our proposed algorithm, the benchmark test StirMark for audio (Steinebach *et al.*, 2001) was undertaken using default parameters (Table 3). The watermarking algorithm has strong resistance to the common attacks. The NC results of detecting watermark using our proposed algorithm were compared with those using the algorithms in Wang and Zhao (2006) and Chen and Zhu (2008). The results of our algorithm are slightly better. As Wang and Zhao (2006) used a similar synchronization method, the NC results against ‘CutSamples’, ‘FFT_Invert’, ‘Invert’, and ‘VoiceRemove’ attacks using our method are similar to those using Wang and Zhao (2006)’s method, but better than those using Chen and Zhu (2008)’s method. Our proposed algorithm, however, is sensitive to such StirMark attacks as ‘CopySamples’, ‘Compressor’, and ‘Amplify’.

Table 3 Comparison results of robustness against the StirMark benchmark for audio (SMBA)

Attack	Normalized correlation coefficient		
	Wang and Zhao (2006)	Chen and Zhu (2008)	Ours
AddBrumm	0.992	0.932	1
AddNoise	0.521	0.763	0.893
AddSinus	0.708	0.906	1
Amplify	0.346	0.549	0.857
Compressor	0.269	0.730	0.822
CopySamples	0.570	0.674	0.765
CutSamples	1	0.532	1
ExtraStereo	1	1	1
FFT_Invert	1	0.849	1
FFT_Real_reverse	0.986	0.980	1
Original	1	1	1
FlippSample	1	0.636	1
Invert	1	1	1
LSBzero	0.956	0.689	0.997
Nothing	1	1	1
Normalize	0.934	0.875	0.986
Stat1	0.318	0.949	1
Stat2	0.902	0.994	1
VoiceRemove	1	0.793	1
ZeroCross	0.676	0.967	0.986

4.3 Watermark payload

Watermark payload was measured as the maximum number of bits per second of watermark message embedded without any extraction error. In other words, it refers to the number of bits that are embedded into the original audio within a unit of time. Suppose the sampling rate of the host audio is F_s (Hz), the length of the host audio is N_s, and the watermark data are of N_w bits. The data payload D_p is defined as follows:

$$D_p = \frac{N_w F_s}{N_s}. \tag{19}$$

Based on this equation, the data payload of our scheme is 32 bits/s, which is comparatively high.

4.4 Security analysis

The security of the embedded watermark should be considered in a watermarking system due to the fact that, if the positions of the embedded watermark are guessed by an attacker successfully, the embedded watermark can be easily detected and altered. In the proposed scheme, the embedding positions are difficult to guess as we adopt a chaotic sequence to shuffle the watermarking position, which can enhance the security of the scheme. Also, the initial values and parameters exploited to generate the chaotic sequence render high security in our scheme. As shown in Fig. 7, with the correct keys, we can extract the watermark correctly as NC is 1. Without correct keys, NC is 0.12 in this case, which means it is impossible to extract the watermark.

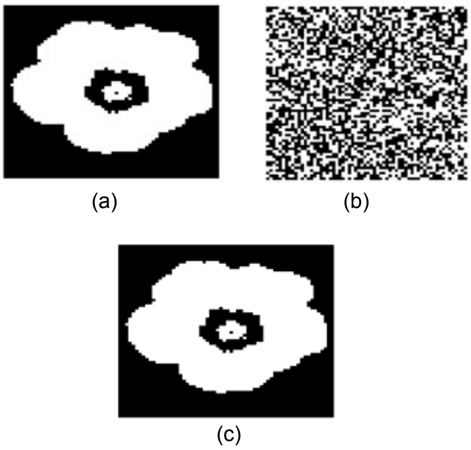

Fig. 7 Security tests

(a) Original watermark; (b) Watermark extracted without correct keys (NC=0.12); (c) Watermark extracted with the correct keys (NC=1)

5 Conclusions

We propose a new audio watermarking technique for content integrity authentication and copyright protection by combining chaotic compound encryption and synchronization techniques. The watermark data are used to modify the most significant DCT coefficients of the corresponding scrambled positions in the host audio for copyright protection and to alter insignificant DCT coefficients for content authentication. Synchronization codes are inserted into the host audio, which leads to robustness to desynchronization attacks. Experimental results show that our watermarking goal is achieved satisfactorily using our multipurpose scheme. The test results also show that the proposed audio watermarking algorithm is robust against common audio processing manipulations and benchmark attacks of the StirMark for audio. Our proposed method achieves better performance than other similar schemes.

References

Chen, N., Zhu, J., 2008. Multipurpose audio watermarking algorithm. *J. Zhejiang Univ.-Sci. A*, **9**(4):517-523. [doi:10.1631/jzus.A071493]

Cvejic, N., Seppanen, T., 2008. Digital Audio Watermarking Techniques and Technologies: Applications and Benchmarks. Information Science Reference, Hershey.

Huang, J., Wang, Y., Shi, Y.Q., 2002. A Blind Audio Watermarking Algorithm with Self-Synchronization. Proc. IEEE Int. Symp. on Circuits and Systems, **3**:627-630.

Katzenbeisser, S., Petitcolas, F.A.P., 2000. Information Hiding Techniques for Steganography and Digital Watermaring. Artech House, Norwood, Mass, USA.

Kirovski, D., Malvar, H.S., 2003. Spread-spectrum watermarking of audio signals. *IEEE Trans. Signal Process.*, **51**(4):1020-1033. [doi:10.1109/TSP.2003.809384]

Lu, C.S., Liao, H.Y.M., 2001. Multipurpose watermarking for image authentication and protection. *IEEE Trans. Image Process.*, **10**(10):1579-1592. [doi:10.1109/83.951542]

Lu, C.S., Liao, H.Y.M., Chen, L.H., 2000. Multipurpose Audio Watermarking. Proc. 15th Int. Conf. on Pattern Recognition, p.282-285. [doi:10.1109/ICPR.2000.903540]

Lu, Z.M., Xu, D.G., Sun, S.H., 2005. Multipurpose image watermarking algorithm based on multistage vector quantization. *IEEE Trans. Image Process.*, **14**(6):822-831. [doi:10.1109/TIP.2005.847324]

Ma, X., Li, X., Liu, W., 2008. A New Audio Watermarking Method for Copyright Protection and Tampering Localization. Proc. 3rd Int. Conf. on Innovative Computing Information and Control, p.352-358.

Steinebach, M., Dittmann, J., Seibel, C., Ferri, L.C., Petitcolas, F.A.P., Fatès, N., Fontaine, C., Raynal, F., 2001. StirMark Benchmark: Audio Watermarking Attacks. Proc. Int. Conf. on Information Technology: Coding and Computing, p.49-54. [doi:10.1109/ITCC.2001.918764]

Swanson, M.D., Zhu, B., Tewfik, A.H., Boney, L., 1998. Robust audio watermarking using perceptual masking. *Signal Process.*, **66**(3):337-355. [doi:10.1016/S0165-1684(98)00014-0]

Tsekeridou, S., Solachidis, V., Nikolaidis, N., Nikolaidis, A., Tefas, A., Pitas, I., 2001. Statistical analysis of a watermarking system based on Bernoulli chaotic sequences. *Signal Process.*, **81**(6):1273-1293. [doi:10.1016/S0165-1684(01)00044-5]

Wang, X.Y., Zhao, H., 2006. A novel synchronization invariant audio watermarking scheme based on DWT and DCT. *IEEE Trans. Signal Process.*, **54**(12):4835-4840. [doi:10.1109/TSP.2006.881258]

Wang, Y., Yang, Y., 2008. A Synchronous Audio Watermarking Algorithm Based on Chaotic Encryption in DCT Domain. Proc. Int. Symp. on Information Science and Engineering, **2**:371-374. [doi:10.1109/ISISE.2008.28]

Wu, Y.T., Shih, F.Y., 2007. Digital watermarking based on chaotic map and reference register. *Pattern Recogn.*, **40**(12):3753-3763. [doi:10.1016/j.patcog.2007.04.013]

Xie, L., Zhang, J.S., He, H.J., 2006. NDFT-Based Audio Watermarking Scheme with High Security. Proc. 18th Int. Conf. on Pattern Recognition, **4**:270-273.

Yeo, I.K., Kim, H.J., 2003. Modified patchwork algorithm: a novel audio watermarking scheme. *IEEE Trans. Speech Audio Process.*, **11**(4):381-386. [doi:10.1109/TSA.2003.812145]

Zhang, L., Chen, L.M., Qian, G.B., 2006. Self-Synchronization Adaptive Blind Audio Watermarking. Proc. 2nd Int. Conf. on Multi-media Modelling, p.1-4.